## Microsoft PLR 4-2, Exhibit A: Mini Markman Preliminary Claim Construction – Global

Below is Microsoft's preliminary construction of the "Virtual Distribution Environment" ("VDE") "invention" of the February 13, 1995, InterTrust application (the "VDE Invention") and certain other terms, which constructions are incorporated by reference into Microsoft's preliminary construction of certain other disputed claims, claim terms, and claim phrases.[1]

| Required Feature | Construction |
|---|---|
| **Security and Commerce World** | InterTrust's February 13, 1995, patent application described as its "invention" a Virtual Distribution Environment ("VDE invention") for **securing**, administering, and **auditing** all **security** and commerce digital information within its multi-node world (community). VDE guarantees to all VDE "participants" identified in the patent application that it will maintain the **availability**, **secrecy**, **integrity** and **authenticity** of all such information present at any appliance (node) within the VDE world (including protected content (including currency, credit, payments, etc.), information about content usage, content-**control information**, **controls**, **load modules**, etc.). VDE is **secure** against at least the threats identified in the patent application to this **availability** (no user may delete the information without **authorization**), **secrecy** (neither available nor disclosed to unauthorized persons or processes), **integrity** (neither intentional nor accidental alteration), and **authenticity** (asserted characteristics are genuine). VDE further provides and requires the components and capabilities described below. Anything less than or different than this is not VDE or the described "invention." |
| **VDE Secure Processing Environment** | At each node where VDE-protected information is **accessed**, **used**, or assigned **control information**, VDE requires a Secure Processing Environment. A Secure Processing Environment is uniquely identifiable, self-contained, non-circumventable, and trusted by all other VDE nodes to protect the **availability**, **secrecy**, **integrity** and **authenticity** of all information identified in the patent application as being protected, and to guarantee that such information will be **accessed** and **used** only as expressly **authorized** by the **associated VDE controls**. A Secure Processing Environment is formed by, and requires, a special-purpose **Secure Processing Unit** having a hardware **tamper-resistant barrier encapsulating** a processor and internal **secure memory**. The **barrier** prevents all **unauthorized** interference, removal, observation, and **use** of the information and processes within it. A Secure Processing Environment is under **control** of **controls** and **control information** provided by one or more parties, rather than being under **control** of the appliance's users or programs. |
| **VDE Controls** | VDE **allows access** to or **use** of protected information and processes only through execution of (and satisfaction of the requirements imposed by) independent, |

[1] The word "invention" is used not to suggest that anything described in InterTrust's patents in fact was novel or non-obvious or inventive, but rather to identify what was described as the **alleged** invention. Also, features and capabilities are described as they are described in the InterTrust patent application, even though the patent application did not describe an actual working system having any of these capabilities. Also, Microsoft's proposed constructions use many terms from the InterTrust patents that are used inconsistently or otherwise indefinitely in the patents. Those terms are used by Microsoft in their narrowest applicable sense, and without waiving the right to assert the indefiniteness of this claim language. Also, the preliminary constructions assume (without conceding) that the February, 1995, InterTrust patent application was incorporated by reference into the '721, '861, and '683 patents, effectively for claim construction purposes. If the Court concludes otherwise, then the proper constructions will be different in some cases. Bolded terms are preliminarily defined in Exhibits A-C of Microsoft's PLR 4-2 papers.

| Required Feature | Construction |
|---|---|
| | special-purpose, executable VDE control(s). A VDE control can execute only within a **Secure Processing Environment**. Each VDE control is a **component assembly** dedicated to a particular activity (e.g., editing, modifying another control, a user-defined action, etc.), particular user(s), and particular protected information. Each separate information **access** or **use** is independently **controlled** by independent VDE control(s). Each VDE control is assembled, within a **Secure Processing Environment**, from independently deliverable modular components (e.g., **load modules** or other controls), dynamically in response to an information **access** or **use request**. The dynamic assembly of a control is directed by a "blueprint" **record** (put in place by one or more VDE users) **containing control information** identifying the exact modular code components to be assembled and executed to **govern** this particular activity on this particular information by this particular user(s). Each control is independently assembled, loaded and delivered vis-à-vis other controls. **Control information** and controls are extensible and can be configured and modified by all users, and combined by all users with any other VDE **control information** or controls (including that provided by other users), subject only to "senior" user controls. Users can assign **control information** and controls to an arbitrarily fine, user-defined portion of the protected information, such as a single paragraph of a document, as opposed to being limited to file-based controls. VDE controls reliably limit use of the protected information to **authorized** activities and amounts. |
| **VDE Secure Containers** | A VDE secure container is a self-contained, self-protecting data structure which (a) encapsulates information of arbitrary size, type, format, and organization, including other, nested, containers, (b) cryptographically protects that information from all **unauthorized access** and **use**, (c) provides encrypted storage management functions for that information, such as hiding the physical storage location(s) of its protected contents, (d) permits the **association** of itself or its contents with **controls** and **control information governing access** to and **use** thereof, and (e) **prevents** such **use** or **access** (as opposed to merely **preventing** decryption) until it is **opened**. A secure container can be **opened** only as expressly **allowed** by the **associated VDE control(s)**, only within a **Secure Processing Environment**, and only through decryption of its encrypted header. A secure container is not directly accessible to any non-VDE calling process. All such calls are intercepted by VDE. The creator of a secure container can assign (or allow others to assign) **control information** to any arbitrary portion of a secure container's contents, or to an empty secure container (to **govern** the addition of contents to the secure container, and **access** to or **use** of those contents). A container is not a secure container merely because its contents are encrypted and signed. A secure container is itself **secure**. All VDE-protected information (including protected content, information about content usage, content-**control information, controls**, and **load modules**) is encapsulated within a secure container whenever stored outside a **Secure Processing Environment** or **secure database**. |
| **Non-Circumventable** | VDE is non-circumventable (sequestered). It intercepts all attempts by any and all users, processes, and devices, to **access** or **use** (e.g., observe, interfere with, or remove) protected information, and prevents all such attempts other than as allowed by execution of (and satisfaction of all requirements imposed by) **associated VDE controls** within **Secure Processing Environment(s)**. |
| **Peer to Peer** | VDE is peer-to-peer. Each VDE node has the innate ability to perform any role identified in the patent application (e.g., end user, content packager, distributor, **clearinghouse**, etc.), and can protect information flowing in any direction between any nodes. VDE is not client-server. It does not pre-designate and restrict one or more nodes to act solely as a "server" (a provider of information (e.g., authored content, **control information**, etc.) to other nodes) or "client" (a requestor of such information). All types of protected-content transactions can proceed without requiring interaction with any server. |
| **Comprehensive Range of** | VDE comprehensively governs all **security** and commerce activities identified in the patent application, including (a) metering, budgeting, monitoring, reporting, and **auditing** information usage, (b) billing and paying for information usage, and (c) negotiating, signing and enforcing contracts that establish users' rights to |

| Required Feature | Construction |
|---|---|
| **Functions** | access or use information. |
| **User-Configurable** | The specific protections governing specific VDE-protected information are specified, modified, and negotiated by VDE's users . For example, VDE enables a consumer to place limits on the nature of content that may be accessed at her node (e.g., no R-rated material) or the amount of money she can spend on viewing certain content, both subject only to other users' senior controls. |
| **General Purpose; Universal** | VDE is universal as opposed to being limited to or requiring any particular type of appliance, information, or commerce model. It is a single, unified standard and environment within which an unlimited range of electronic rights protection, data security, electronic currency, and banking applications can run. |
| **Flexible** | VDE is more flexible than traditional information security and commerce systems. For example, VDE allows consumers to pay for only the user-defined portion of information that the user actually uses, and to pay only in proportion to any quantifiable VDE event (e.g., for only the number of paragraphs displayed from a book). |